

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-120044

(43)公開日 平成11年(1999) 4月30日

(51)Int.Cl.<sup>6</sup>

G 0 6 F 12/00

識別記号

5 0 1

5 2 0

F I

G 0 6 F 12/00

5 0 1 B

5 2 0 P

審査請求 有 請求項の数5 O L (全 13 頁)

(21)出願番号 特願平9-285454

(22)出願日 平成9年(1997)10月17日

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72)発明者 佐々 哲

東京都品川区北品川6丁目7番35号 ソニ

ー株式会社内

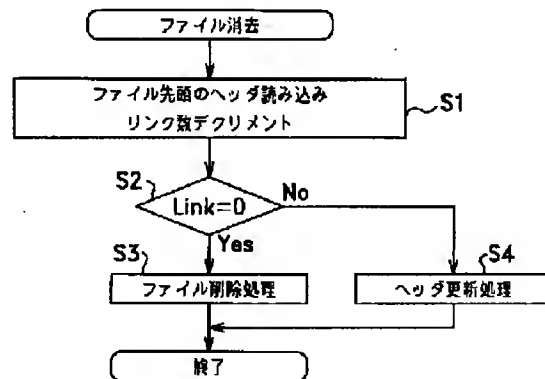
(74)代理人 弁理士 小池 晃 (外2名)

(54)【発明の名称】 データ処理装置、データ処理方法、データ処理システム及び記録媒体

(57)【要約】

【課題】 他のデータとの参照関係を有するデータを削除しても当該他のデータの参照関係に影響が生じるのを防止する。

【解決手段】 CPUは、消去の対象となっているファイルのヘッダ部(512バイト)をフラッシュメモリから読み込んでRAMに格納する。そして、OSヘッダ内にある「リンク数」を1つ減らして、さらに、消去の対象となっているファイルの属するディレクトリにおいて、そのディレクトリファイルのシステムエントリエリアから消去対象となっているファイルに関するディレクトリエントリを削除する(ステップS1)。CPUは、新たな「リンク数」が零であるかを判定し、零になったと判定したときはこのファイルを削除し(ステップS3)、零になっていないと判定したときはOSヘッダを更新して新たな「リンク数」とする(ステップS4)。



ファイルを削除するときのCPUの動作を説明するフローチャート

## 【特許請求の範囲】

【請求項1】 記録媒体に記録されているデータの処理を行うデータ処理装置において、

上記記録媒体に記録された各データ毎の参照先のデータを示す参照情報と参照されているデータの数を示すリンク情報とを記憶する記憶手段と、

上記記憶手段に記憶されている参照情報と上記リンク情報とに基づいて上記記録媒体における各データの参照関係を管理する管理手段とを備え、

上記管理手段は、上記記録媒体に記録されている一のデータの参照先のデータを削除するときは、削除対象となるデータに対応したリンク情報の示す数を1つ減らして新たなリンク情報を算出し、新たなリンク情報の示す数が零になった場合は上記削除対象のデータを上記記録媒体から削除し、新たなリンク情報の示す数が零でない場合は上記削除対象となるデータのリンク情報を上記新たなリンク情報になるように更新して上記一のデータの参照情報が示す参照先から上記削除対象となるデータを削除することを特徴とするデータ処理装置。

【請求項2】 記録媒体に記録されているデータを管理するデータ処理方法において、

上記記録媒体に記録された各データ毎の参照先のデータを示す参照情報と参照されているデータの数を示すリンク情報とを記憶し、

記憶されている参照情報と上記リンク情報とに基づいて上記記録媒体における各データの参照関係を管理し、

上記記録媒体に記録されている一のデータの参照先のデータを削除するときは、削除対象となるデータに対応するリンク情報の示す数を1つ減らして新たなリンク情報を算出し、

新たなリンク情報の示す数が零になった場合は上記削除対象のデータを上記記録媒体から削除し、新たなリンク情報の示す数が零でない場合は上記削除対象となるデータのリンク情報を上記新たなリンク情報になるように更新して上記一のデータの参照情報が示す参照先から上記削除対象となるデータを削除することを特徴とするデータ処理方法。

【請求項3】 参照先のデータを示す参照情報と参照されているデータの数を示すリンク情報とを有する複数のデータを記録する記録媒体と、

上記記録媒体に記録された各データ毎の参照情報とリンク情報とを記憶する記憶手段と、上記記憶手段に記憶されている参照情報と上記リンク情報とに基づいて上記記録媒体における各データの参照関係を管理する管理手段とを有する管理部とを備え、

上記管理部の管理手段は、上記記録媒体に記録されている一のデータの参照先のデータを削除するときは、削除対象となるデータに対応したリンク情報の示す数を1つ減らして新たなリンク情報を算出し、新たなリンク情報の示す数が零になった場合は上記削除対象のデータを上

記記録媒体から削除し、新たなリンク情報の示す数が零でない場合は上記記録媒体の上記削除対象となるデータのリンク情報を上記新たなリンク情報になるように更新して上記一のデータの参照情報が示す参照先から上記削除対象となるデータを削除することを特徴とするデータ処理システム。

【請求項4】 参照先のデータを示す参照情報と参照されているデータの数を示すリンク情報とを有する複数のデータを記録し、

各データ毎の参照情報とリンク情報とを上記記録媒体から読み出して記憶し、

記憶されている参照情報と上記リンク情報とに基づいて上記記憶された各データの参照関係を管理し、

記憶されている一のデータの参照先のデータを削除するときは、削除対象となるデータに対応したリンク情報の示す数を1つ減らして新たなリンク情報を算出し、

新たなリンク情報の示す数が零になった場合は上記削除対象のデータを削除し、新たなリンク情報の示す数が零でない場合は上記削除対象となるデータのリンク情報を上記新たなリンク情報になるように更新して上記一のデータの参照情報が示す参照先から上記削除対象となるデータを削除することを特徴とするデータ処理方法。

【請求項5】 参照先のデータを示す参照情報と参照されているデータの数を示すリンク情報とを有する複数のデータを記録することを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、記録媒体に記録されているデータを管理するデータ処理装置等に関する。

【0002】

【従来の技術】光ディスク、磁気ディスク、フラッシュメモリ等の記録媒体に記憶されているデータは、例えばホストコンピュータがこれらのデータを管理するための情報を上記記録媒体から読み出して階層ディレクトリ構造を構築することによって、管理される。また、上記記録媒体に記録されているデータは、ディレクトリ構造構築のための情報だけでなく、参照先のデータを示す情報も備えている。これにより、ホストコンピュータは、ディレクトリ構造によってデータを管理するだけでなく、データとデータの参照関係を認識することによっても管理することが可能である。

【0003】

【発明が解決しようとする課題】しかし、ホストコンピュータは、参照関係の情報を有するデータを上記記録媒体から読み出した場合、参照関係に問題を生じることなくかかるデータの処理を行うことが難しかった。

【0004】例えば、参照関係の有無に関係なくデータを削除してしまうと、このデータを参照していたデータは、参照先がなくなってしまい、また、削除されたデータと同じ番号やファイル名が他のデータに用いられた場

合には誤ったデータを参照してしまう問題が生じる。

【0005】このような参照関係を有するものの例として、一のディレクトリに存在するファイルを他のディレクトリからもアクセスするようにするために、上記ファイルと上記他のディレクトリとをリンクするリンク機能がある。このとき、上記ファイルを上記一のディレクトリから削除した場合に、上記他のディレクトリのリンクを無視してしまうと、上記他のディレクトリはファイルにアクセス不可能になるだけでなく、異なるファイルにアクセスする問題が生じる。

【0006】このような問題に対処すべく、ファイルを削除する際に、このファイルにリンクしている全てのディレクトリ又はファイルを調べてリンクを変更する処理が考えられる。しかし、全てのディレクトリ等を調べる処理は非常に困難であり、処理が煩雑になるおそれがある。単にリンクを削除する処理も同様に、全てのリンク先のファイル等を調べる必要があり、処理負担が非常に大きい。

【0007】本発明は、このような実情に鑑みて提案されたものであり、他のデータとの参照関係を有するデータを削除しても当該他のデータの参照関係に影響が生じるのを防止するデータ処理装置、データ処理方法、データ処理システム及び記録媒体を提供することを目的とする。

【0008】

【課題を解決するための手段】上述の課題を解決するために、本発明に係るデータ処理装置は、記録媒体に記録されているデータの処理を行うデータ処理装置において、上記記録媒体に記録された各データ毎の参照先のデータを示す参照情報と参照されているデータの数を示すリンク情報とを記憶する記憶手段と、上記記憶手段に記憶されている参照情報と上記リンク情報とに基づいて上記記録媒体における各データの参照関係を管理する管理手段とを備え、上記管理手段は、上記記録媒体に記録されている一のデータの参照先のデータを削除するときは、削除対象となるデータに対応したリンク情報の示す数を1つ減らして新たなリンク情報を算出し、新たなリンク情報の示す数が零になった場合は上記削除対象のデータを上記記録媒体から削除し、新たなリンク情報の示す数が零でない場合は上記削除対象となるデータのリンク情報を上記新たなリンク情報になるように更新して上記一のデータの参照情報が示す参照先から上記削除対象となるデータを削除することを特徴としている。

【0009】本発明に係るデータ処理方法は、記録媒体に記録されているデータを管理するデータ処理方法において、上記記録媒体に記録された各データ毎の参照先のデータを示す参照情報と参照されているデータの数を示すリンク情報とを記憶し、記憶されている参照情報と上記リンク情報とに基づいて上記記録媒体における各データの参照関係を管理し、上記記録媒体に記録されている

一のデータの参照先のデータを削除するときは、削除対象となるデータに対応するリンク情報の示す数を1つ減らして新たなリンク情報を算出し、新たなリンク情報の示す数が零になった場合は上記削除対象のデータを上記記録媒体から削除し、新たなリンク情報の示す数が零でない場合は上記削除対象となるデータのリンク情報を上記新たなリンク情報になるように更新して上記一のデータの参照情報が示す参照先から上記削除対象となるデータを削除することを特徴としている。

- 10 【0010】本発明に係るデータ処理システムは、参照先のデータを示す参照情報と参照されているデータの数を示すリンク情報とを有する複数のデータを記録する記録媒体と、上記記録媒体に記録された各データ毎の参照情報とリンク情報とを記憶する記憶手段と、上記記憶手段に記憶されている参照情報と上記リンク情報とに基づいて上記記録媒体における各データの参照関係を管理する管理手段とを有する管理部とを備え、上記管理部の管理手段は、上記記録媒体に記録されている一のデータの参照先のデータを削除するときは、削除対象となるデータに対応したリンク情報の示す数を1つ減らして新たなリンク情報を算出し、新たなリンク情報の示す数が零になった場合は上記削除対象のデータを上記記録媒体から削除し、新たなリンク情報の示す数が零でない場合は上記記録媒体の上記削除対象となるデータのリンク情報を上記新たなリンク情報になるように更新して上記一のデータの参照情報が示す参照先から上記削除対象となるデータを削除することを特徴としている。

- 20 【0011】本発明に係るデータ処理方法は、参照先のデータを示す参照情報と参照されているデータの数を示すリンク情報とを有する複数のデータを記録し、各データ毎の参照情報とリンク情報とを上記記録媒体から読み出して記憶し、記憶されている参照情報と上記リンク情報とに基づいて上記記憶された各データの参照関係を管理し、記憶されている一のデータの参照先のデータを削除するときは、削除対象となるデータに対応したリンク情報の示す数を1つ減らして新たなリンク情報を算出し、新たなリンク情報の示す数が零になった場合は上記削除対象のデータを削除し、新たなリンク情報の示す数が零でない場合は上記削除対象となるデータのリンク情報を上記新たなリンク情報になるように更新して上記一のデータの参照情報が示す参照先から上記削除対象となるデータを削除することを特徴としている。

- 30 【0012】本発明に係る記録媒体は、参照先のデータを示す参照情報と参照されているデータの数を示すリンク情報とを有する複数のデータを記録することを特徴としている。

【0013】

【発明の実施の形態】以下、本発明の実施の形態について、図面を参照しながら説明する。

- 40 【0014】本発明は、図1に示す構成のメモ리카ード

システム1に適用される。メモリカードシステム1は、ホストコンピュータ10とメモリカード20により構成される。

【0015】上記ホストコンピュータ1は、具体的には図1に示すように、静止画像や音声等の様々なファイルや各ファイルをまとめて管理するディレクトリ等を記憶するハードディスク11と、ハードディスク11等からのファイル等を一旦記憶して読み出すRAM(Random Access Memory)12と、3本のラインを介してメモリカード20とのデータの送受信を行う第1のシリアルインターフェース(以下、「第1のシリアルI/F」という。)13と、各回路を制御するCPU(Central Processing Unit)14とを備え、例えばハードディスク11に記憶されている画像や音声等のファイルをディレクトリを用いた階層構造によって管理することができる。

【0016】RAM12は、例えばハードディスク11に記憶されているファイルを一旦記憶し、必要に応じてこのファイルをバスを介して第1のシリアルI/F13に供給する。

【0017】第1のシリアルI/F13は、3本のラインを介して、メモリカード20にデータを送信したり、メモリカード20に記憶されているデータを受信する。具体的には、第1のシリアルI/F13は、第1のラインを介して、上記制御データやファイルの送信の際のシリアルクロックSCKを送信する。第1のシリアルI/F13は、第2のラインを介して、第1のラインでのファイル又は制御データ等のシリアルデータの切り換えに応じて、そのときの状態を示すチップセレクト信号CSを出力する。さらに、第1のシリアルI/F13は、第3のラインを介して、メモリカード20に書き込むためのファイルや制御データを送信したり、メモリカード20から読み出されたファイルを受信する。

【0018】CPU14は、RAM12やハードディスク11のファイルを読み出したり、RAM12等にファイルを書き込むことを制御したり、メモリカード20とファイル等の送受信の制御も行う。例えば、CPU14は、メモリカード20の図示しない誤消去防止スイッチのライトプロテクトがオンになっているかを判定するためのレジスタ命令を発行したり、メモリカード20に対してアドレスを指定して所定のファイルの書き込み命令を発行する。

【0019】ここで、メモリカード20は、図1に示すように、上記ホストコンピュータ1からのファイルや制御データをシリアルで送受信する第2のシリアルI/F21と、ファイルやディレクトリを記憶するフラッシュメモリ22と、フラッシュメモリ22に記憶されているファイル等の読出し又は書き込みを制御するコントローラ23とを備える。

【0020】上記フラッシュメモリ22は、例えばNAND型のものであり、図2に示すように、ホストコンピ

ュータ1から送信されたファイルが消去単位となるブロック(例えば8kバイト又は16kバイト)毎に分割して記憶される。各ブロック0, 1, 2, ..., nは、ファイル等を512バイト記憶するデータエリアと16バイトの冗長エリアとによってなる1ページ(=512バイト+16バイト)が集合して構成される。上記冗長エリアには、当該ブロックのデータエリアにあるデータを管理する分散管理情報が記憶されている。なお、これらの分散管理情報が集合すると集合管理情報(集合管理ファイル)が構成される。集合管理ファイルは、数ブロックで構成されて、全てのブロックのデータの状態を一括して管理するものである。

【0021】ここで、ファイルは、画像や音声等に用いるユーザファイル形式のものと、システム管理のためのシステムファイル形式のものとがあり、フラッシュメモリ22の1以上のブロックのデータエリアにおいて記憶される。

【0022】ユーザファイル形式のファイルは、図3に示すように、当該ファイルの属性情報である512バイトのヘッダ部と主情報であるデータ部によって構成される。ヘッダ部は、当該ファイルをメモリカード20内で管理するための情報である128バイトのOSヘッダと、汎用的で必要不可欠な管理情報である240バイトのファイルヘッダと、144バイトのエントリエリアとからなる。また、データ部は、テーブルデータ領域と、4つのエントリデータとを有する。なお、上述の図3に示すデータ部では、4つまでのエントリデータが格納される。すなわち、ユーザファイル形式のファイルは、最大で4つのエントリデータから構成される。また、図3では、エントリデータとして、インデックス、画像データ、2つの補助データを例に挙げたが、特に限定されるものではなく、その他音声データ等であってもよい。

【0023】上記OSヘッダは、具体的には図4に示すように、「ファイルID」(2バイト)、「ファイルバージョン」(2バイト)、「ファイルサイズ」(4バイト)、「使用ブロック数」(2バイト)、「リンク数」(1バイト)、「日付」(8バイト)、「メーカー/機種コード」(4バイト)、「初期登録ディレクトリ番号」(2バイト)、「アプリケーションカテゴリー」(1バイト)、「ファイル名」(11バイト)、「キーワード登録数」(1バイト)、「キーワード文字コード」(1バイト)、「キーワード文字列」(32バイト)、「リセットリザーブ」(4バイト)、「個別データ」(32バイト)、さらに未使用領域「リザーブ」を有する。

【0024】「ファイルID」は、ファイルの種別(用途)を表すものであり、ファイルヘッダにも同様のものがある。「ファイルバージョン」は、いわゆるバージョンナンバーを示すものであり、ホストコンピュータ10やメモリカード20においてこのファイルを処理するこ

とができるかを判定するために用いられる。「ファイルサイズ」は、ヘッダ部及びデータ部からなる当該ファイルの全体の大きさをバイト数で示すものである。「使用ブロック数」は、フラッシュメモリ22で当該ファイルを使用しているブロック数を示すものである。「リンク数」は、当該ファイルを参照しているディレクトリの数を示すものである。なお、「リンク数」には、当該ファイルの属するディレクトリも含まれる。例えば、他のディレクトリから参照されていないファイルは、その属するディレクトリのみとリンクしているので、「リンク数」は1である。「日付」は、ファイルを作成した日付又は更新した日付を示すものである。「メーカー／種別コード」は、メモリカード20にファイルを書き込んだ機器のメーカー名及びその機種を示すものである。「初期登録ディレクトリ番号」は、最初に登録したディレクトリ番号を示すものであり、ファイルが別のディレクトリに移った場合でも更新されるものではない。「アプリケーションカテゴリー」は、このファイルが用いられるアプリケーションのカテゴリーを示すものである。「ファイル名」は、ホストコンピュータ10でファイルを管理するときに用いられるファイル名であり、メモリカード20内では特に使用されない。「キーワード登録数」は、登録されたキーワードの数を示すものである。「キーワード文字コード」は、キーワードの文字コードを示したものである。「キーワード文字列」は、キーワード、セパレータを決めて複数指定をすることもできるのである。「0リセットリザーブ」は、書換えの際に必ず0にしておくものである。「個別データ」は、ファイルID毎に管理するためのもので、例えばファイルIDによって決められる用途で用いられるデータであり、利用者が自由に用いることができる。

【0025】上記ファイルヘッダは、具体的には図5に示すように、「規格識別データ」(8バイト)、「ファイル規格識別データ」(8バイト)、「ファイルID」(2バイト)、「ファイルバージョン」(2バイト)、「アプリケーション作成日付」(8バイト)、「アプリケーション更新日付」(8バイト)、「作成メーカー／機種コード」(4バイト)、「更新メーカー／機種コード」(4バイト)、「0リセットリザーブ」(16バイト)、「データエントリ数」(1バイト)、「テーブル数」(1バイト)、「文字コード」(1バイト)、「タイトル文字列」(128バイト)、さらに未使用領域「リザーブ」を有する。

【0026】「規格識別データ」は、当該フラッシュメモリ22が所定の規格に沿ってファイルを記憶していることを示すものである。「ファイル規格識別データ」は、当該ファイルが上記所定の規格に沿って作成されたことを示すものである。「ファイルID」は、ファイルの種別を表すものであり、OSヘッダにも同様のものがある。「ファイルバージョン」は、バージョンナンバー

を示すものである。「アプリケーション作成日付」は、アプリケーションを作成した日付を示すものであり、「アプリケーション更新日付」は、それを更新した日付を示すものである。「作成メーカー／種別コード」は、ファイルを作成したメーカー及びその機種名を示すものであり、「更新メーカー／種別コード」は、そのファイルを更新したメーカー及びその機種名を示すものである。「データエントリ数」は、後述のエントリデータの数を示すものである。「テーブル数」は、テーブルデータ領域のデータ数を示すものである。「文字コード」は、入力文字を所定のコード番号で示したものである。「タイトル文字列」は、タイトル文字を示したものである。

【0027】エントリエリアは、データ部の4つのエントリデータを管理するためのデータ(以下、「エントリ」という。)がそれぞれ記憶されている。

【0028】上記エントリエリア内の各エントリは、4つのエントリデータに対応して設けられ、図5に示すように、4つのエントリデータ毎に、「開始アドレス」(4バイト)、「データサイズ」(4バイト)、「データ種別ID」(1バイト)、「リザーブ」(1バイト)、「個別データ」(26バイト)がある。例えば、エントリ1の「開始アドレス」は図3に示す「エントリデータ1 インデックス」の先頭論理アドレスを示し、エントリ2の「開始アドレス」は「エントリデータ2 画像データ」の先頭論理アドレスを示す。

【0029】一方、システムファイル形式のファイルとしては、例えばディレクトリファイルが該当し、ユーザファイル形式のものと比べて、ヘッダ部の構成が異なっている。

【0030】ここで、1つのディレクトリは、1つのファイル、すなわち1つのディレクトリファイルによって構成され、最大8kバイトの大きさとする。ディレクトリファイルは、ヘッダ部とデータ部とから構成され、主にヘッダ部のディレクトリインフォメーションとデータ部のディレクトリエントリリストとによって各ファイルを管理する。

【0031】上記ディレクトリファイルのヘッダ部は、図6に示すように、「OSヘッダ」(128バイト)、「ファイルヘッダ」(240バイト)、「システムエントリエリア」(48バイト)、「ディレクトリインフォメーション」(96バイト)を有する。なお、「OSヘッダ」と「ファイルヘッダ」は、上述したユーザファイル形式のものと同様である。

【0032】「システムエントリエリア」は、ユーザファイル形式のものとほとんど同じであるが、各エントリに個別データが設けられていない。具体的には図7に示すように、各エントリ毎に、「開始アドレス」(4バイト)、「データサイズ」(4バイト)、「データ種別ID」(4バイト)がある。例えば一のエントリの「開始

10

20

30

40

50

アドレス」は後述のグループインフォメーションの先頭論理アドレスを示し、他のエントリの「開始アドレス」は後述のディレクトリエントリの先頭論理アドレスを示す。

【0033】「ディレクトリインフォメーション」は、ディレクトリの情報を管理するものであり、図8に示すように、「ディレクトリID」(2バイト)、「対象ファイルID」(1バイト)、「グループ数」(1バイト)、「親ディレクトリ」(2バイト)、「ディレクトリ数」(2バイト)、「ファイル数(全ファイル)」(2バイト)、「対象ファイル数」(2バイト)、「アクセスポイント」(2バイト)、「ディレクトリエントリ数」(2バイト)、「システム用ディレクトリエントリ数」(2バイト)、「使用システムディレクトリエントリ数」(2バイト)、「確保ブロック数」(2バイト)、「使用ブロック数」(2バイト)、さらに未使用領域「リザーブ」(72バイト)を有する。

【0034】「ディレクトリID」はディレクトリの種別を表し、「対象ファイルID」はそのディレクトリで高速にアクセスするファイル(対象ファイル)のIDを示すものである。「グループ数」は登録されたグループの数である。「親ディレクトリ」は、親ディレクトリ、すなわち1つ上の階層のファイル番号を示すものである。「ディレクトリ数」は含まれるディレクトリの総数を示し、「ファイル数(全ファイル)」は当該ディレクトリに含まれる全ファイルの数を示す。「対象ファイル数」は含まれる対象ファイル数を示し、「アクセスポイント」はアクセス開始ディレクトリエントリを示すものである。「ディレクトリエントリ数」は使用ユーザディレクトリエントリ数を示し、「システム用ディレクトリエントリ数」は確保システム用のディレクトリエントリ数を示す。「使用システムディレクトリエントリ数」はシステム用のディレクトリエントリ数を示し、「確保ブロック数」はアプリケーションで確保したいフラッシュメモリ22のブロック数を示し、「使用ブロック数」はアプリケーションで使用したブロック数を示す。

【0035】また、上記ディレクトリファイルのデータ部は、上述の図6に示すように、エントリデータ1として「グループインフォメーション」(可変長)、エントリデータ2として「ディレクトリエントリリスト」(可変長)を有する。

【0036】「グループインフォメーション」は、図9に示すように、「グループ番号」(1バイト)、「グループインフォメーションサイズ」(1バイト)、「ディレクトリID」(2バイト)、「0リセットリザーブ」(1バイト)、「リザーブ」(3バイト)、「日付」(8バイト)、「個別データ」(8バイト)、「タイトル」(可変長)とを有する。

【0037】「グループインフォメーション」は、ディレクトリ内で所望のファイルのグルーピングを行うため

の情報であり、詳しくは後述するが、実質的にはディレクトリの下階層に疑似ディレクトリを設けるものである。「グループインフォメーションサイズ」は、グループインフォメーションの大きさを示すものであり、固定長24バイト+「タイトル」の任意バイト数の大きさになる。「ディレクトリID」はディレクトリ種別毎に付けられるIDであり、「0リセットリザーブ」は修正する場合は必ず0に設定するものである。「リザーブ」は当該ファイル作成時に0に設定するものである。「日付」は当該ファイルの作成日付を示し、これは使用者により変更可能である。「個別データ」は特に定められたものでなく、「タイトル」はディレクトリファイルのタイトルを示すものである。

【0038】「ディレクトリエントリリスト」は、ディレクトリに属する要素を示すものであり、図10に示すように、ディレクトリエントリが複数集合して構成される。なお、1つのディレクトリエントリは4バイトである。ディレクトリエントリは、「ファイル番号」(2バイト)、「属性情報」(1バイト)、「グループ番号」(1バイト)で構成される。

【0039】「ファイル番号」は、図11に示すように、2バイト中の最初のビットには「0」があり、残りの15ビットにおいてファイル又はディレクトリの先頭論理アドレスが示されている。なお、「0x f f f f」が示されている時は、未使用を意味する。

【0040】「属性情報」は、図12に示すように、「ディレクトリ/ファイル」(1ビット)、「対象フラグ」(1ビット)、「マーク1」(1ビット)、「マーク2」(1ビット)、「連続記録」(2ビット)、「リザーブ」(2ビット)の情報を有する。「ディレクトリ/ファイル」は、1のときは当該ディレクトリの下にさらにディレクトリがあることを示し、0のときはファイルがあることを示す。「対象フラグ」は、「1」のときは上述のディレクトリインフォメーションで指定した対象ファイルIDのファイルであることを示し、「0」のときはそれ以外のファイルであることを示す。すなわち、「対象フラグ」が立っている(1)ときは、「ファイル番号」によって指定されるファイルを当該ディレクトリにおいて高速にアクセスすることができることを意味する。なお、「対象フラグ」が立っていない(0)ときはアプリケーション毎に取扱いが異なる。「マーク1」は、1のときはマーク1指定のあることを示し、0のときはかかる指定がないことを示す。「マーク2」については、「マーク1」と同様である。なお、「マーク1」及び「マーク2」は、利用者が指定できるマーキングフラグである。「連続記録」は、00のときは通常のファイル、11のときは先頭の連続ファイル、10のときは連続指定されたファイルであることを示す。

【0041】「グループ番号」は、図13に示すように、8ビットで示され、1階層の疑似階層を示すグルー

ブ番号である。これにより、同じディレクトリ内で同じグループ番号を有するファイルは、1つのグループとして取り扱われる。なお、グループ番号0のファイルは、何もグルーピングされていないものであり、疑似的にディレクトリに見せかけたときは上記ディレクトリの直下のものとなる。

【0042】ここで、同一のディレクトリ内に、ファイル1～ファイル9の9つのファイルが存在していたとする。さらに、ファイル1、5、7はグループ1に属し、ファイル2、4はグループ2に属し、ファイル3、6はグループ3に属し、ファイル8、9はグループ0に属するものとする。

【0043】このとき、グループ1～グループ3は、図14に示すように、上記ディレクトリの下の階層に位置し、疑似的にディレクトリが存在しているような役割を果たす。但し、グループの下に更にディレクトリを設けることはできない。また、グループ0に属するファイル8、9は、上記ディレクトリの直下のものとして取り扱われる。

【0044】なお、例えば図15に示すように、各グループは、疑似ディレクトリのように取り扱わなくてもよく、単にグループ関係のあるファイルを示すものとして取り扱ってもよい。

【0045】以上のように構成されるメモ리카ードシステム1において、ホストコンピュータ10のCPU14は、メモ리카ード20のフラッシュメモリ22から論理アドレス0のルートディレクトリのディレクトリファイル、その他のディレクトリファイル、さらに、必要なファイルのヘッダ部を読み出して、各ファイルを管理する。

【0046】具体的には、CPU14は、全ファイルのヘッダ部におけるファイルヘッダからタイトル、日付等の情報を読み出し、エントリエリアの各エントリからディレクトリエントリエリアの開始位置、データサイズ等を読み出し、さらに、ディレクトリファイルのシステムインフォメーション、システムエントリリスト等に基づいて、ルートディレクトリと基幹とする階層ディレクトリ構造を構成する。そして、CPU14は、例えば図16に示すように、全ファイル及び全ディレクトリの階層構造を形成することができる。

【0047】図16において、論理アドレス0のルートディレクトリは、基幹として最も上の階層に位置し、他のディレクトリや複数のファイルを管理する。ルートディレクトリの下の階層に位置するディレクトリも、他のディレクトリや複数のファイルを管理する。

【0048】また、各ファイルは、1つのディレクトリに属しているだけでなく、他のディレクトリから参照される関係を有することがある。

【0049】例えば図17に示すように、ファイル1～ファイル4がディレクトリ1に属し、ファイル5はディ

レクトリ2に属し、ファイル6はディレクトリ3に属しているものとする。このとき、ディレクトリ2はファイル3、4を参照し、ディレクトリ3はファイル4、5を参照しているものとする。

【0050】従来では、例えばディレクトリ1からファイル3を削除すると、ディレクトリ2は、ファイル3を参照することができなくなったり、参照先が他のファイルに代わってしまうこともあった。そこで、各ファイルは、OSヘッダに「リンク数」を設けてディレクトリとリンクしている数の情報を有し、リンクされているファイルを削除しても不都合が生じないようにしている。具体的には、ホストコンピュータ10のCPU14は、ファイルを削除する場合では、図18に示すステップS1～ステップS4の処理を行う。

【0051】ステップS1において、CPU14は、消去の対象となっているファイルのヘッダ部(512バイト)をフラッシュメモリ22から読み込んでRAM12に格納する。そして、OSヘッダ内にある「リンク数」(例えばnとする。)を1つ減らして(n-1)として、さらに、消去の対象となっているファイルの属するディレクトリにおいて、そのディレクトリファイルのディレクトリエントリリストから消去対象となっているファイルに関するディレクトリエントリを削除して、ステップS2に進む。

【0052】ステップS2において、CPU14は、新たな「リンク数」、すなわち(n-1)が零であるかを判定し、零になったと判定したときはステップS3に進み、零になっていないと判定したときはステップS4に進む。

【0053】ステップS3において、CPU14は、消去の対象となっているファイルをフラッシュメモリ22から消去する処理を行う。

【0054】一方、ステップS2において零になっていないと判定したときのステップS4において、CPU14は、(n-1)を新たな「リンク数」とするようにOSヘッダを更新して、フラッシュメモリ22に書き込む。

【0055】これにより、例えば図17に示すファイル2は、ディレクトリ1から削除される対象になった場合が、ディレクトリ1からファイル2に関するディレクトリエントリが削除されるとともに、ステップS1の処理によって「リンク数」が1から0になり、ステップS2、ステップS3の処理を介して、図19に示すように、実際に削除される。

【0056】一方、ファイル3は、ディレクトリ1から削除される対象となった場合は、ディレクトリ1からファイル3に関するディレクトリエントリが削除されるとともに、ステップS1の処理によって「リンク数」が2から1になる。ファイル3は、ディレクトリ2からのリンクは削除されていないので、ステップS2、ステップ



S4の処理を介して、図19に示すように、削除されずに「リンク数」だけ減らされる。なお、ファイル5についても同様である。

【0057】つぎに、ディレクトリを介してファイルとファイルが容易にリンクされることについて、図20を参照しながら説明する。ここでは、ルートディレクトリを基幹としてカメラディレクトリ、プリントディレクトリが形成されている場合を例に挙げる。

【0058】図20において、カメラディレクトリの下  
の階層には、画像1ファイル、画像2ファイル、画像3  
ファイル、画像4ファイル、画像5ファイルが設けられ  
ている。また、プリントディレクトリの下の階層には、  
例えばプリントアウトするときのサイズや枚数等の情報  
が記録されているプリントファイルが設けられている。

【0059】ここで、プリントディレクトリは、上記図  
10に示すディレクトリエントリリスト内に、プリント  
ファイルを示すディレクトリエントリだけでなく、画像  
1ディレクトリエントリ、画像2ディレクトリエント  
リ、画像3ディレクトリエントリを有する。画像1ディ  
レクトリエントリは、画像1ファイルのファイル番号や  
属性情報等を備え、画像2ディレクトリエントリは、画  
像2ファイルのファイル番号や属性情報等を備える。画  
像3ディレクトリエントリ、画像4ディレクトリエント  
リも同様である。また、このプリントディレクトリにお  
いて、プリントファイルを示すディレクトリエントリで  
は、「対象フラグ」は立てられており、画像1ファイル  
～画像4ファイルに対応するディレクトリエントリで  
は、「対象フラグ」は立てられていない。

【0060】これにより、プリントディレクトリは、プ  
rintファイルに高速にアクセスすることができ、また  
、画像1ファイル～画像4ファイルとリンクしてい  
る。換言すると、プリントファイルは、プリントディ  
レクトリを介して、画像1ファイル～画像4ファイルとリ  
nkしている。このとき、プリントファイルが実行され  
ると、プリントディレクトリを介して、画像1ファイル  
～画像4ファイルの画像が所定の大きさ・枚数でプリ  
ntアウトされるようになっている。

【0061】つぎに、例えば画像2ファイルをカメラデ  
ィレクトリから削除する場合について説明する。画像2  
ファイルは、カメラディレクトリとプリントディレクト  
リにリンクしている。したがって、上述のステップS1  
において「リンク数」は「2」から「1」になる。そし  
て、ステップS2、ステップS4によって、カメラディ  
レクトリから画像2ファイルに関するディレクトリエン  
トリが削除されるが、画像2ファイルはヘッダが更新さ  
れるだけで実際には削除されない。これにより、プリ  
ntファイルは、アルバム1ディレクトリから画像2ファ  
イルが削除されても、画像2ファイルをプリントアウト  
することができる。

【0062】また、プリントディレクトリは、グループ

機能を使うことによって、複数のプリントファイルを所  
有することができる。図21では、プリントディレク  
トリの下の階層に、グループ1、グループ2が設けられ  
ている。ここでは、図20に示すプリントファイルのディ  
レクトリエントリと画像1～画像4のディレクトリエン  
トリとをグループ1とし、プリントファイル2のディレ  
クトリエントリと画像5ディレクトリエントリとをグル  
ープ2としている。これにより、プリントファイル1を  
実行すれば、画像1ファイル～画像4ファイルの画像を  
プリントアウトすることができ、プリントファイル2を  
実行すれば、画像5ファイルの画像をプリントアウトす  
ることができ、プリントアウトしたい画像をグループ分  
けすることが可能になる。

【0063】以上のように、ディレクトリを介すること  
によって、ファイルとファイルを容易にリンクすること  
が可能になる。すなわち、一のファイルと他のファイル  
をリンクする場合は、上記一のファイルの属するディレ  
クトリのディレクトリエントリに、上記他のファイルの  
「ファイル番号」等を登録するだけでよい。

【0064】これに対して、ファイルとファイルを直接  
リンクしようとする、そのファイルの中を調べてリン  
ク先のデータを調べる必要がある。これでは、ファイル  
の種類が増えれば増えるほど、その分全てのファイルの  
構造を把握しなければならず、ファイルの中を調べるた  
めの処理負担が大きくなる。特に、アプリケーションが  
増えるほど異なる構造のファイルが増える傾向があるの  
で、ホストコンピュータ10の処理負担が大きくなる。

【0065】しかし、上述のように、ディレクトリを介  
してファイルとファイルをリンクすると、ファイルの種  
類が増大しても、リンク元であるファイルのディレクト  
リのディレクトリエントリを調べることによって、容易  
にリンク先を調べることができる。

【0066】また、ファイルをディレクトリから削除し  
ようとする場合において、そのファイルが他のディレク  
トリから参照されているときは実際には削除されないの  
で、リンクしているディレクトリからそのファイルにア  
クセスすることができる。すなわち、この場合は、実際  
には削除されないで、ディレクトリのリンク先が異な  
るファイルに化けるのを回避することができる。また、  
ファイルを削除する際には、このファイルの「リンク  
数」のみを調べているので、このファイルにリンクして  
いる全てのファイルを調べる必要がなくなり、従来に比  
べて処理負担を大幅に減らすことができる。

【0067】なお、本実施の形態では、記録媒体の一例  
としてフラッシュメモリを備えるメモリカードを例に挙  
げて説明したが、本発明はこれに限定されるものではな  
く、例えば磁気ディスクや光ディスク等の媒体にも適用  
可能である。

【0068】

【発明の効果】以上詳細に説明したように、本発明に係



るデータ処理装置及びデータ処理方法によれば、記録媒体に記憶されている一のデータの参照先のデータを削除するときは、削除対象となるデータの有するリンク情報の示す数を1つ減らして新たなリンク情報を算出し、新たなリンク情報の示す数が零になった場合は削除対象のデータを記録媒体から削除し、新たなリンク情報の示す数が零でない場合は削除対象となるデータのリンク情報を新たなリンク情報になるように更新して一のデータの参照情報から削除対象となるデータを削除することにより、参照先のデータを削除する場合であってもこのデータが複数のデータと参照関係を有するときは実際に削除するのを防止して、他のデータとの参照関係を維持することができる。

【0069】本発明に係るデータ処理システム及びデータ処理方法によれば、制御部の制御手段は、記録媒体に記憶されている一のデータの参照先のデータを削除するときは、削除対象となるデータの有するリンク情報の示す数を1つ減らして新たなリンク情報を算出し、新たなリンク情報の示す数が零になった場合は削除対象のデータを記録媒体から削除し、新たなリンク情報の示す数が零でない場合は記録媒体の削除対象となるデータのリンク情報を新たなリンク情報になるように更新して一のデータの参照情報から削除対象となるデータを削除することによって、参照先のデータを削除する場合であってもこのデータが複数のデータと参照関係を有するときは実際に削除するのを防止して、他のデータとの参照関係を維持することができる。

【0070】本発明に係る記録媒体によれば、参照先のデータを示す参照情報と参照されている数を示すリンク情報とを有する複数のデータを記憶することにより、参照情報とリンク情報とに基づいて上記複数のデータの参照関係を構築することができる。

#### 【図面の簡単な説明】

【図1】本発明を適用したメモ리카ードシステムの構成を示すブロック図である。

【図2】上記メモ리카ードシステムのメモ리카ード内のフラッシュメモリに格納されるデータの構成を説明するための図である。

【図3】上記フラッシュメモリに格納されるファイルの構成を説明するための図である。

【図4】上記ファイルのOSヘッダの構成を説明するた

めの図である。

【図5】上記ファイルのファイルヘッダ及びエントリエリアの構成を説明するための図である。

【図6】ディレクトリファイルの構成を説明するための図である。

【図7】上記ディレクトリファイルのシステムエントリエリアの構成を説明するための図である。

【図8】上記ディレクトリファイルのディレクトリインフォメーションの構成を説明するための図である。

10 【図9】上記ディレクトリファイルのグループインフォメーションの構成を説明するための図である。

【図10】上記ディレクトリファイルのディレクトリエントリリストの構成を説明するための図である。

【図11】上記ディレクトリエントリリストのファイル番号を説明するための図である。

【図12】上記ディレクトリエントリリストの属性情報を説明するための図である。

【図13】上記ディレクトリエントリリストのグループ番号を説明するための図である。

20 【図14】ディレクトリ内のグループの取り扱いを説明するための図である。

【図15】ディレクトリ内のグループの取り扱いを説明するための図である。

【図16】ディレクトリとファイルによって構成されるディレクトリ階層を説明するための図である。

【図17】ディレクトリとファイルの参照関係を説明する図である。

【図18】ファイルを削除するときのCPUの動作を説明するフローチャートである。

30 【図19】ディレクトリとファイルの参照関係を説明する図である。

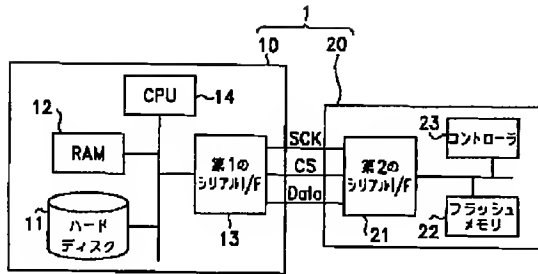
【図20】プリントディレクトリと画像ファイルの参照関係を示す図である。

【図21】プリントディレクトリと画像ファイルの参照関係を示す図である。

#### 【符号の説明】

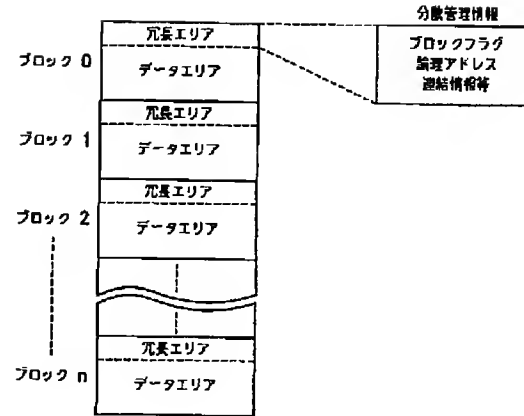
1 メモ리카ードシステム、10 ホストコンピュータ、11 ハードディスク、12 RAM、13 第1のシリアルI/F、14 CPU、15 20メモ리카ード、21 第2のシリアルI/F、22 フラッシュメモリ、23 コントローラ

【図1】

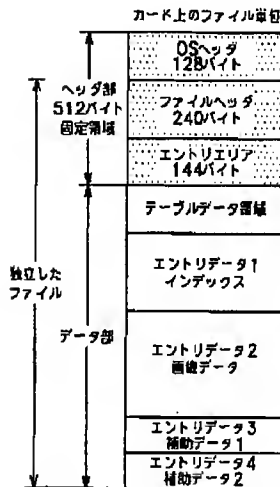


本発明を適用したメモリカードシステムの構成を示すブロック図

【図2】



【図3】



ユーザファイル形式のファイルの構成を示す図

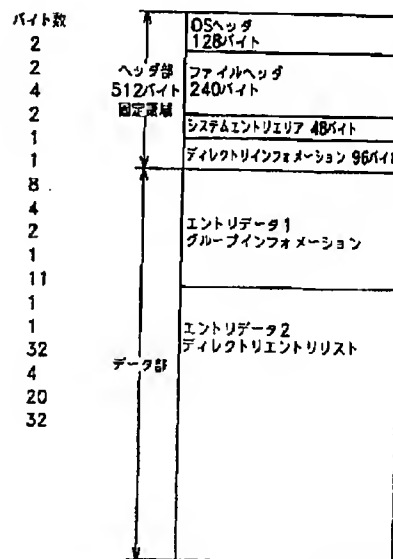
フラッシュメモリの名ブロックに記憶されるデータの説明図

【図4】

OSヘッダ	
ファイルID	2
ファイルバージョン	2
ファイルサイズ	4
使用ブロック数	2
リンク数	1
システムリザーブ	1
日付	8
メーカー/機種コード	4
初期登録ディレクトリ番号	2
アプリケーションカテゴリー	1
ファイル名	11
キーワード登録数	1
キーワード文字コード	1
キーワード文字列	32
0リセットリザーブ	4
リザーブ	20
個別データ	32
ファイルID等に使用	

OSヘッダの構成を示す図

【図6】



ディレクトリファイルの構成を示す図

【図8】

ディレクトリID	2Byte
対象ファイルID	1Byte
グループ数	1Byte
親ディレクトリ	2Byte
ディレクトリ数	2Byte
ファイル数(全ファイル)	2Byte
対象ファイル数	2Byte
アクセスポイント	2Byte
ディレクトリエントリ数	2Byte
システム用Dirエントリ数	2Byte
使用システムDirエントリ数	2Byte
確保ブロック数	2Byte
使用ブロック数	2Byte
リザーブ	72Byte

ディレクトリリンク情報の構成を示す図

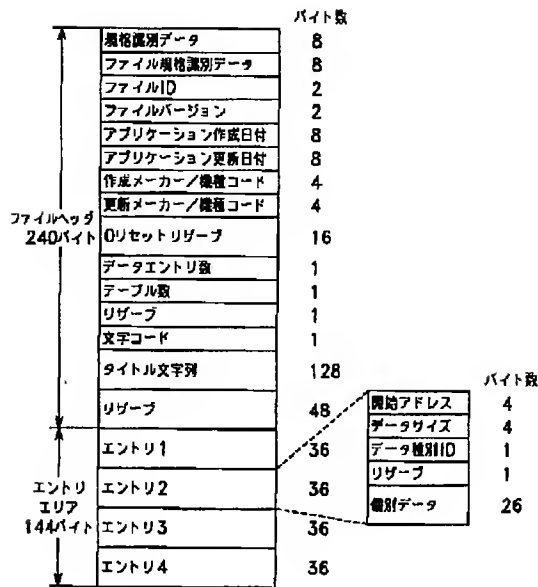
【図11】

ファイル番号(2バイト)

0	ファイル番号(15bit)
ファイル番号	ファイルあるいはディレクトリファイルの先頭 管理アドレス 0xffff 未使用

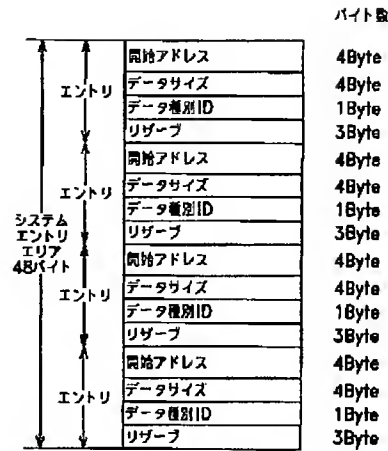
各ディレクトリエントリのファイル番号を示す図

【図5】



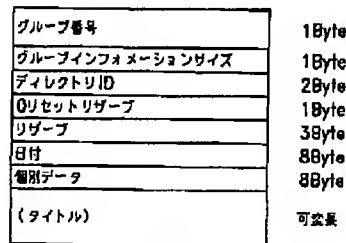
ファイルヘッダ及びエントリエリアの構成を示す図

【図7】



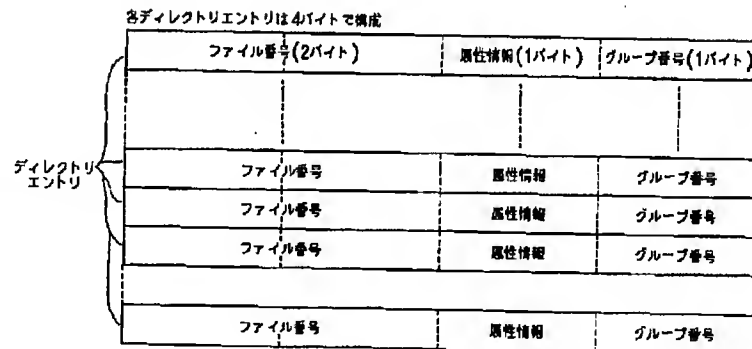
システムエントリエリアの構成を示す図

【図9】



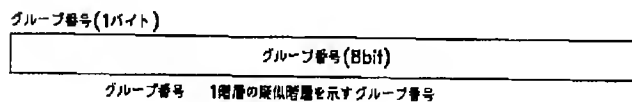
グループインフォメーションの構成を示す図

【図10】



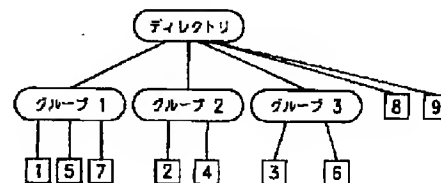
ディレクトリエントリリストの構成を示す図

【図13】



各ディレクトリエントリのグループ番号を示す図

【図14】



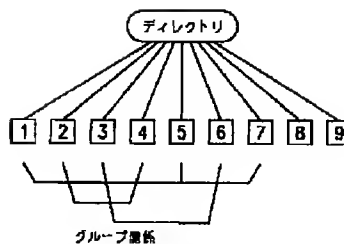
グループ分けによる疑似ディレクトリの説明図

【図12】

属性情報(1バイト)				
Dir/File	対象フラグ	マーク1	マーク2	連続記録
Dir/File	1: Dir 0: File			
対象フラグ	1: ディレクトリインフォメーションで指定した対象ファイルIDのファイル 0: それ以外のファイル			
マーク1	1: マーク1指定あり(ユーザが指定できるマーキングフラグ) 0: なし			
マーク2	1: マーク2指定あり(ユーザが指定できるマーキングフラグ) 0: なし			
連続記録	00: 通常のファイル 11: 先頭の連続ファイル 10: 連続指定されたファイル			
リザーブ	00: 初期値			

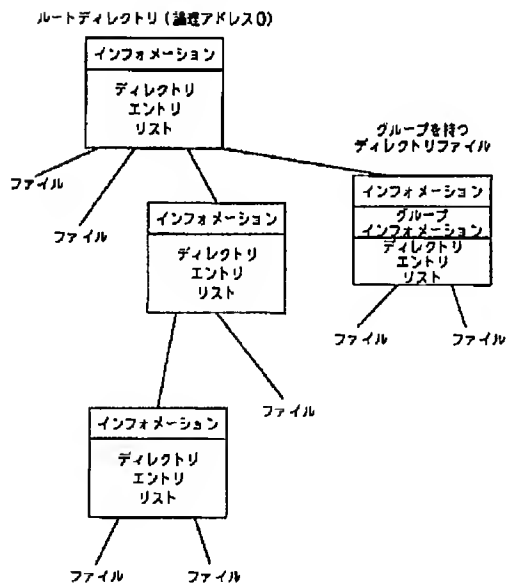
各ディレクトリエントリの属性情報の構成を示す図

【図15】



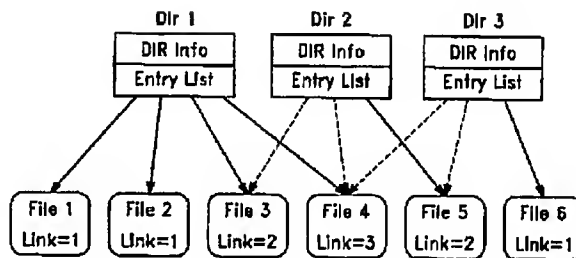
複数ディレクトリに見えない場合のディレクトリとファイルの関係を示す図

【図16】



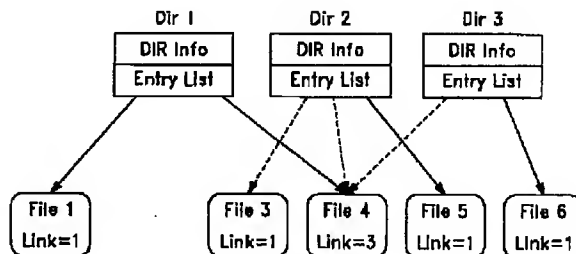
ディレクトリとファイルによる階層構造を説明する図

【図17】



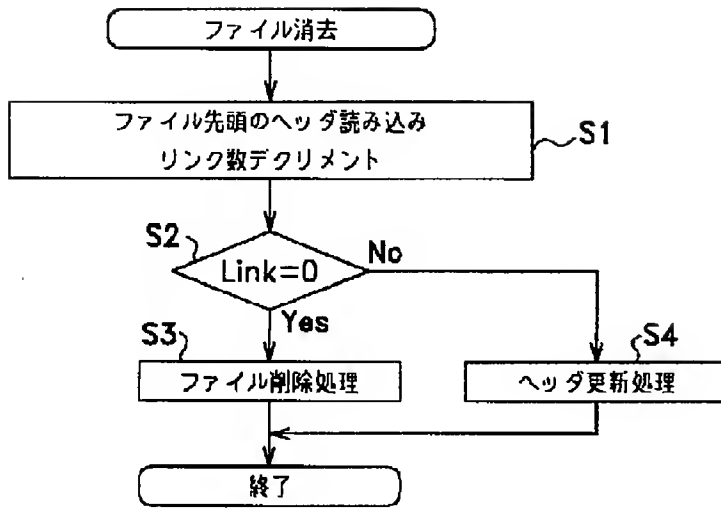
ディレクトリとファイルの参照関係の説明図

【図19】



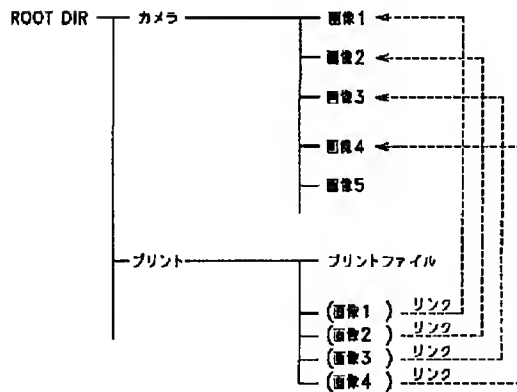
ディレクトリとファイルの参照関係の説明図

【図18】



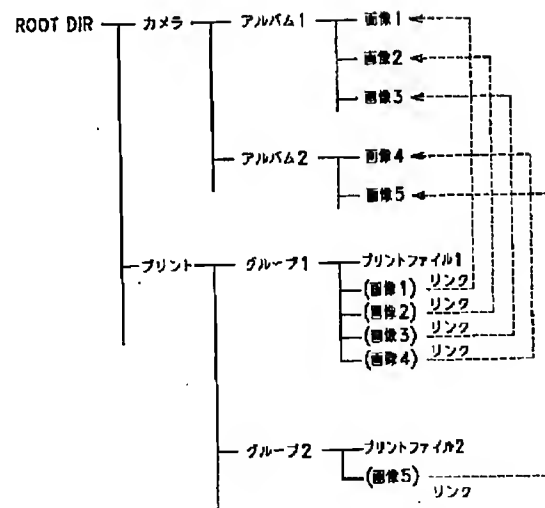
ファイルを削除するときのCPUの動作を説明するフローチャート

【図20】



プリントディレクトリと画像ファイルの参照関係を示す図

【図21】



プリントディレクトリと画像ファイルの参照関係を示す図